

SYSTEM AND METHOD FOR COOPERATIVE
EDITING OF WEB DOCUMENT

Field of the Invention

5

This invention relates to a system and a method for cooperative editing of a web document using a personal computer (PC), wherein a plurality of editors are coupled to a network, and to a computer-readable record media to store instructions for performing the method; and, more particularly, to a system and a method for cooperative editing of a web document by producing a cooperative editing session having a formation of client/server.

15

Description of the prior Art

An apparatus for cooperative editing of a web document is designed hierarchically, divided into a T.120 protocol stack, the web document controller and a user interface. The T.120 protocol stack serves a data transmission and a production and control of cooperative editing session. The web document controller serves to process editing instructions and synchronize the instructions, so that all editors can keep a copy of the same web document. The user interface serves to produce the editing instructions and display a modified web document.

There are some conventional methods for cooperative

editing. The conventional methods include an application program sharing or a shared white board, et al. By the method for cooperative editing with an application program sharing, the application program on the display is shared but only one editor is endowed with an authority to edit. By the method for cooperative editing with a shared white board, an authority is open to all editors, so that text or image can be added/deleted in the cooperative area. By the former method, all the types of media can be edited, however, cooperative editing is performed using a function of sharing the application program, such that the editors cannot work for editing at the same time. By the latter method, it is impossible to edit a text document. Thus, there is a problem that these two methods do not provide enough function for cooperative editing.

Summary of the Invention

It is an object of the present invention to provide a system and a method for cooperative editing a web document, wherein a plurality of editors can edit the web document cooperatively at the same time by executing the system for cooperative editing the web document on each terminal coupled to a network.

It is another object of the present invention to provide a computer-readable record media performing the method.

In accordance with an aspect of the present invention,

there is provided, in a cooperative computer based system for editing a web document including a plurality of clients and a server, the cooperative computer based system including: a) means for generating editing instructions, converting data format of editing instructions to the data format for a message related to editing and transmitting the message related to editing to the server; b) means for determining if the web document can be modified, based on the message related to editing; modifying contents of a copied web document stored in the server in conform to the message related to editing; and transmitting the message related to editing to the plurality of the clients; and c) means for modifying contents of the copied web document stored in each of the plurality of the clients in conform to the message related to editing transmitted from the server.

In accordance with another aspect of the present invention, there is provided, in a cooperative computer based system for editing a web document including a server and a plurality of clients, a computer-readable record media storing instructions for performing a method for editing a web document cooperatively, comprising the steps of: a) at a client among the plurality of the clients, generating editing instructions, converting data format of editing instructions to the data format for a message related to editing and transmitting the message related to editing to the server; b) at the server, determining if the web document can be modified, based on the message related to editing; modifying contents of

a copied web document stored in the server in conform to the message related to editing; and transmitting the message related to editing to the plurality of the clients; and c) modifying contents of the copied web document stored in each 5 of the plurality of the clients in conform to the message related to editing transmitted from the server.

Brief Description of the Drawings

10 Other objects and aspects of the invention will become apparent from the following description of the embodiments with reference to the accompanying drawings, in which:

15 Fig. 1 is a pictorial representation of a structure of a cooperative editing system in accordance with the present invention;

Fig. 2A is a pictorial representation of a window when executing a cooperative editing apparatus;

Fig. 2B is another pictorial representation of a window when executing a cooperative editing apparatus;

20 Fig. 3 is a pictorial representation in a window when executing a cooperative editing apparatus;

Fig. 4 is a schematic diagram illustrating a relationship between a cooperative editing apparatus and T.120 protocol stack;

25 Fig. 5 is a block diagram illustrating a cooperative apparatus for editing a web document in accordance with the present invention;

Fig. 6 is a flow chart representing a process of commencing cooperative editing according to a method for cooperative editing of a web document in accordance with the present invention;

5 Fig. 7 is a data structure representing a line in case a web document is converted into an internal form;

Fig. 8 is a block diagram representing a process of processing editing instructions according to a method for a web document editing in accordance with the present invention;

10 Fig. 9A is a flow chart representing a process of processing editing instructions at a client according to a method for a web document editing in accordance with the present invention;

Fig. 9B is a flow chart representing a process of 15 processing an editing message at a client according to a method for a web document editing in accordance with the present invention; and

Fig. 10 is a flow chart representing a process of 20 processing an editing message at a server according to a method for a web document editing in accordance with the present invention.

Detailed Description of the Preferred Embodiments .

25 As shown in Fig. 1, one server must be in one session, in accordance with a definition that all functions operate in the environment of client/server. A server receives all editing

messages generated in the cooperative editing session and distributes them to editors. A computer in which a web document to be edited is opened first and a session is generated, determines if a server token is in the session. If 5 the server token is available, the computer becomes a server by acquiring the server token.

Generally, the server serves as a server until the session ends. However, in case of changing servers of the session, the server token is handed over to another client. A 10 new server taking over the server token, informs all computers in the session that it has become the new server. Since the server is distinguished from the client by having the server token, it does not happen that more than one server are in one session.

15 Each editor has an authority to editing within an area assigned to it and no authority beyond the area. Generally, web site managers and web designers, et al., have the authority within wide area assigned to them. On the contrary, sponsors and contents providers have the authority within a 20 little area. A session manager controls the authority to editing, that is, the session manager has the authority to editing within all area of a web document and can assign the authority to the editors. The session manager makes, deletes, and stores files and manages users in the cooperative editing 25 session.

As shown in Fig. 2A, a window of a cooperative editing apparatus is divided into two windows. One is used for WYSIWYG

editing window 21 for editing in the WYSIWYG form, and the other is used for text editing window 22 for editing in the text form. Editing is performed on each window, and when contents are edited on the WYSIWYG editing window, the same 5 edited contents are maintained on the text-editing window. In Fig. 3, a reference numeral 23 denotes the editor's caret used to edit the text.

Referring to Fig. 2B, when executing the cooperative editing apparatus, two windows independently controlled, are 10 generated. For example, a reference numeral 25 denotes a window displaying a web document text and a reference numeral 24 denotes a window displaying a result of rendering the web document. The web document text window 25 is the same as the text-editing window 22. Two windows 25 and 24 are executed by 15 two methods.

One is a method that, as can be seen in Fig. 2B, both a WYSIWIG editing and a text editing can be executed as a single application by executing a WYSIWYG window built-in in the cooperative editing apparatus.

20 The other is a method that only a result of editing is informed by executing a general web browser. At this time, the WYSIWIG editing is impossible but the text editing is possible. And each time contents of a web document being edited are modified, it is required to inform the web browser of this 25 modification. To do so, it is required to store the web document being edited in the web document form and transmit a message to the web browser, so that the web browser can read

this file. At this time, in case of editing a web site directly, it is required to convert a uniform resource locator (URL) included in the web document to an absolute address form commencing with http://, so that the web browser can bring and read pictures from the web site directly. In case of cooperatively editing a web document stored in a local disk, it is required to copy to provide all picture files to a client, so that the client can edit the web document by providing the same environment as the local disk of the server.

10 Naturally, it is possible to use a method of editing the web document on the two windows and then informing the web browser of contents by using two methods mentioned above together.

In Fig. 3, a reference numeral 31 denotes an editor's caret and a reference numeral 32 an area which an editor 15 requests. This area 32 is protected from the other editors until the editor or the session manager releases a Lock on the area. Since the editor's authority to editing is limited by line, the editor can do lock and edit the limited lines allowed to be edited. The editor distinguishes the area of his 20 locking from the area of the other editor's locking by a color difference between his text and the other's. A reference numeral 33 denotes a line number used for discerning between each line.

In accordance with the present invention, one instant number is given to one line and the server controls this instant number. When a line is added, the server gives this line a new instant number. When a line is deleted, also a

corresponding instant number is deleted. When a line is modified, a corresponding number is not changed.

Referring to Fig. 4, before executing the cooperative editing apparatus, T.120 protocol stack must be executed, 5 because an application program 45 of T.120 operates the cooperative editing apparatus. The T.120 protocol stack for a multimedia data conference includes a multipoint communication service (MCS) 41, a generic conference control (GCC) 42 and a node controller (NC) 43. The MCS 41 transmits data and 10 controls a resource. The GCC 42 controls a conference and a session. The NC 43 generates, maintains and controls the conference by using the T.120 protocol stack. The MCS 41 and the GCC 42 respectively have a standard of T.122/125 and T.124, however, the NC 43 has no standard. A white board, a file 15 transmission, an application program sharing and a chatting are application programs 45 usable during the conference and these programs have a standard of T.126, T.127, T.128 and T.134 respectively.

A standard APE 44 is a method for transmitting data 20 between peer application programs, and has a standard of T.121. The web document cooperative editing apparatus 47 uses a non-standard APE 46, in order to exchange data between apparatus. Modifying the standard APE 44 generates the non-standard APE 25 46 suitable for cooperative editing. Also, while executing the web document cooperative editing apparatus, T.120 application programs, for example, of a file transmission, a chatting and a white board are usable, such that it is possible to transmit

a file for editing and to exchange an opinion about editing by using these application programs.

Referring to Fig. 5, the web document cooperative editing apparatus 50 includes a user interface module 51, a web document manager module 52 and a user manager module 53. The user interface module 51 visualizes the web document by parsing the web document and provides an editing environment of WYSIWYG form. The web document manager module 52 performs a plurality of functions related to editing. The user manager module 53 controls information of users participating in editing, when the web document cooperative editing apparatus operates as a server.

In more detail, the web document manager module 52 includes the functions of: converting the read web document to an internal form for cooperative editing; analyzing an editing message upon arrival of the editing message and then determining if editing instructions are correct, and performing editing; converting editing instructions to message form in order to transmit the editing instructions to the other editors; and informing the user interface module 51, when contents of the web document are modified.

The user manager module 53 also determines if a user is allowed to access by inspecting his/her ID and password when the user logs in.

Referring to Fig. 6, it is required to execute the T.120 protocol stack and generate a conference before executing the web document cooperative editing apparatus. In some case,

there might be proceeding a plurality of conferences in one computer, and thus it is required to select a conference out of those conferences to thereby participate in the conference at step 601. More concretely describing, since the presently proceeding conferences are distinguished by each name, the editor can participate in the conference by selecting a name of a desired conference.

After selecting the conference, a web document to be edited is selected at step 602. That is, the desired web document is selected in list of web documents available for cooperative editing. And then, the user logs in at step 603. In the cooperative editing, log in serves to distinguish a session manager from a general editor and determine if an editor has an authority to editing the web document.

Next, determine if a session for the web document to be edited was already generated at step 604.

If the session was not yet generated, the user generates the session by becoming a server after grabbing a token at step 605. In order to grab the token, the web document cooperative editing apparatus needs to request the T.120 MCS of the token. After loading the web document at step 606, the logic flow proceeds to step 607 to start cooperative editing.

If the session was already generated, the logic flow proceeds to the step 607 to start cooperative editing after receiving the web document from the server as a client at step 608. Upon receiving the web document from the server, information of the web document location, editors' authority

to editing and a Lock is transmitted. For, locating the web document brings pictures, et al., from a uniform resource locator (URL) in the web document. In case of receiving the web document stored in the hard disk, the editor receives 5 picture files, et al., included in the web document and stores them in the editor's local disk.

Referring to Fig. 7, the web document cooperative editing apparatus converts the web document to a doubly linked list for managing it. One node in the list is configured to 10 represent one line of the web document and information necessary for editing is stored in each node.

Hereinafter, a process of converting the web document to the internal form will be described. The process happens only in the server, and clients receive the web document, a form of 15 which was converted into the list in the server. By using the web document converted into the list, all editors participating in the session can maintain the same contents.

In order to perform as above, after reading the web document and dividing it by line, the apparatus stores 20 contents of the line in a contents field 74 of the node.

Second, after numbering each line, the apparatus stores a number assigned to the line in the line number field 71. Each line is distinguished by each number in corresponding line number fields and is maintained as the same to all clients 25 participating in the cooperative editing.

Third, user ID fields 76 of editors having authority to editing this node are stored in the form of list. A place of

storage is a user ID list field 75. A user unregistered in the user ID list field 75 cannot edit this node. In order to register in the user ID list field 75, the unregistered user needs to request the session manager of registration. The 5 request can be made before editing or while editing. In case this list is in a NULL state, all editors have authority to editing.

Fourth, a previous field 72 and a next field 73 contain value of a previous node and a next node, respectively.

10 Fifth, the other fields not described above are set to initial value of "0".

Upon completing the node list, cooperative editing proceeds. If a certain editor edits, a user ID of the editor who edited this node is stored in the user ID field 76 of 15 corresponding node. If this field has a value of "0", contents of a file have never been modified since the file was read for the first time.

There is a case that an editor intending to edit this node requests Lock, so that the other editors cannot edit. In 20 this case, the user ID of the editor who requested the lock is stored in a reserved user ID field 77. Ones who can request the lock are just editors registered in the user ID list field 75 and a manager. If the lock request is released, the reserved user ID field 77 has a value of "0".

25 Referring to Fig. 8, supposing that a client 82 edited a portion of the web document, a user interface module 82b inspects a keyboard input or a mouse input of the editor,

converts the inputs through the mouse or keyboard to editing instructions and then transmits them to a web document manager module 82a. At this time, the user interface module 82b does not show modified contents on the display.

5 The web document manager module 82a determines if the received instructions are correct, and if correct, converts the instructions to a message and transmits a server 81 the converted message. The web document manager module 81A in the server 81 receives the converted message, and determines if
10 the editor has an authority to editing and if the other editor requested Lock. If it is confirmed that the editor has the authority, the web document manager module 81B modifies contents of the web document and informs a user interface module 81a that the contents were modified. Then, the user
15 interface module 81A of the server 81 shows the contents on the display in accordance with the modified contents.

Continuously, the web document manager module 81B informs the user interface module 81A of the contents modification and converts the modified contents to a message and transmits a converted message to clients 82 and 83. After receiving the converted message from the web document manager module 81b of the server 81, document manager modules 82a and 83A of the clients 82 and 83 modify their copies of the web document respectively in accordance with the received message, and
20 informs user interface modules 82b and 83B that contents were modified. Then, finally, the client 82 performs showing the modified contents on its display.

In this case, there is a possibility of a little time delay between the client's editing and confirming actually the edited contents. There is a difference in the time delay, depending on a state of network and a hardware performance of 5 server/client system.

Referring to Fig. 9A, in accordance with the present invention, editing instructions of the web document includes a modification, an insert, a deletion, a division and a combination.

10 When a user orders an instruction for modifying in order to edit a specific area in the web document 901, this instruction is transmitted to a web document manager module. At this time, a line number, modified contents and the user ID are transmitted. After receiving this instruction, the web 15 document manager module determines if modifying is possible 902. If the others performed Lock or the user has no authority to editing, this instruction will not be performed. If modifying is possible, the client converts the received instruction to a message at step 903 and transmits the converted to the server at step 904 and gets in a stand-by mode at step 905. If modifying is not possible, the client 20 gets in the stand-by mode immediately.

Referring to Fig. 9B, if the client receives a modified message including line number, modified contents and a user ID, 25 it compares the user ID included in the message with My User ID at step 912. Here, the My User ID denotes the ID of the user him/herself and is used to be compared with the user ID

contained in the modified message. If the user ID is not equal to My User ID, the client determines if a node of the same line number is in the contents list that the client retains at step 913.

5 Continuously, after modifying contents of the node at step 914, the client informs the user interface module that the contents were modified at step 915. That is, the user interface can modify even contents of its own editing, only when an edited message is received from the server. There are
10 two lists that the user manages. One is an editing list, in which he stores the contents of his editing, in sequence and performs a function of undoing. And the other one is a contents list. The contents list is a doubly linked list of the nodes described in Fig. 7 and systems of all editors hold
15 the same contents in their contents lists.

A way of processing the other instructions is similar to that of modifying. In case of inserting, the client searches a node of the same line number in the list and inserts a node next to the searched node. In case of deletion, the client
20 searches a node of the same line number, and deletes the searched node.

If the user ID is the same as My User ID, it means that the message is the one that the editor itself transmits. In this case, after adding these modified contents to the editing
25 list at step 916, the logic flow proceeds to the step 913 of searching a node of the same line number in the contents list.

As shown in the Fig. 10, a server handles a message

received from a client. When receiving a modified message including a line number, modified contents and a user ID from the client at step 101, the server searches a corresponding node in the contents list at step 102.

5 And, it determines if this node is in a state of Lock at step 103. If the node is not in the state of the lock, the server determines if the user has an authority to editing at step 104, and if not, the server does not handle the modified message, otherwise the server modifies the contents at step 105 and transmits the modified contents to the client at step 106. Then, the user interface is informed of the modified contents, and the contents are shown on the display at step 107.

10
15 If the node is in the state of the lock, the server determines if the user, who requests modification, requested the lock at step 108, and if not, the server does not handle the modified message, otherwise the logic flow proceeds to the step 104 where the server determines if the user has an authority to editing.

20 Although the preferred embodiments of the invention have been disclosed for illustrative purpose, those skilled in the art will appreciate that various modifications, additions, and substitutions are possible, without departing from the scope and spirit of the invention as disclosed in the accompanying 25 claims.